

## CLAIMS

### What is claimed is:

1. A method of computing comprising:
  - reading a data processing representation having code sections with code statements of at least a first and a second programming language;
  - recognizing a first code section with at least code statements of a first programming language;
  - invoking a first code statement processing unit of the first programming language to process the first code section;
  - recognizing a second code section with at least code statements of a second programming language;
  - invoking a second code statement processing unit of the second programming language to process the second code section.
2. The method of claim 1, wherein the first and second code sections are non-overlapping code sections.
3. The method of claim 1, wherein said second code section is embedded within said first code section.
4. The method of claim 1, wherein said first language is a directive language, and said second language is a selected one of XML and Java.
5. The method of claim 1, wherein said first language is Java, and said second language is XML.
6. The method of claim 1, wherein the method further comprises
  - recognizing a third code section with at least code statements of a third programming language;
  - invoking a third code statement processing unit of the third programming language to process the third code section.

7. The method of claim 6, wherein said third code section is embedded within said second code section, and said second code section is embedded within said first code section.
8. The method of claim 6, wherein said first language is a directive language, said second language is Java and said third language is XML.
9. The method of claim 1, wherein the method further comprises recognizing an invocation of a library function within at least a selected one of said first and second code sections; invoking the library function, and outputting the result of the invocation.
10. The method of claim 9, wherein the library function is a selected one of an emit function for outputting execution results, a pop function for returning an element, and a push function for backing up an insertion point.
11. The method of claim 1, wherein the method further comprises recognizing a header section of a selected one of the first and the second programming language; recognizing a directive statement within the header section, enumerating one or more data packages; and importing the enumerated one or more data packages for use within code sections with at least statements of the selected first and second programming language.
12. The method of claim 1, wherein the method further comprises recognizing a header section of a selected one of the first and the second programming language; recognizing a declare statement within the header section, enumerating one or more processing methods; and

instantiating the enumerated one or more processing methods for use within code sections with at least statements of the selected first and second programming language.

13. The method of claim 1, wherein the method further comprises  
recognizing a header section of a selected one of the first and the second programming language;  
recognizing a declare statement within the header section, enumerating one or more instance variables; and  
instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language.
14. A method of computing comprising:  
reading a data processing representation having code sections with code statements of at least a first and a second programming language;  
recognizing a header section of a selected one of the first and the second programming language;  
recognizing a directive statement within the header section, enumerating one or more data packages; and  
importing the enumerated one or more data packages for use by code sections within code sections with at least statements of the selected first and second programming language.
15. The method of claim 14, wherein the method further comprises  
recognizing a declare statement within the header section, enumerating one or more processing methods; and  
instantiating the enumerated one or more processing methods for use within code sections with at least statements of the selected first and second programming language.

16. The method of claim 14, wherein the method further comprises  
recognizing a declare statement within the header section, enumerating  
one or more instance variables; and  
instantiating the enumerated one or more instance variables for use  
within code sections with at least statements of the selected first and second  
programming language.
17. A method of computing comprising:  
reading a data processing representation having code sections with  
code statements of at least a first and a second programming language;  
recognizing a header section of a selected one of the first and the  
second programming language;  
recognizing a first declare statement within the header section,  
enumerating one or more processing methods; and  
instantiating the enumerated one or more processing methods for use  
within code sections with at least statements of the selected first and second  
programming language.
18. The method of claim 17, wherein the method further comprises  
recognizing a second declare statement within the header section,  
enumerating one or more instance variables; and  
instantiating the enumerated one or more instance variables for use  
within code sections with at least statements of the selected first and second  
programming language.
19. A method of computing comprising:  
reading a data processing representation having code sections with  
code statements of at least a first and a second programming language;  
recognizing a header section of a selected one of the first and the  
second programming language;  
recognizing a declare statement within the header section, enumerating  
one or more instance variables; and

instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language.

20. An apparatus comprising:
  - at least one storage unit having stored thereon programming instructions designed to enable the apparatus to
    - read a data processing representation having code sections with code statements of at least a first and a second programming language,
    - recognize a first code section with code statements of at least the first programming language,
    - invoking a first code statement processing unit of the first programming language to process the first code section,
    - recognize a second code section with code statements of at least the second programming language,
    - invoking a second code statement processing unit of the second programming language to process the second code section; and
    - at least one processor coupled to said at least one storage unit to execute said programming instructions.
21. The apparatus of claim 20, wherein the first and second code sections are non-overlapping code sections.
22. The apparatus of claim 20, wherein said second code section is embedded within said first code section.
23. The apparatus of claim 20, wherein said first language is a directive language, and said second language is a selected one of XML and Java.
24. The apparatus of claim 20, wherein said first language is Java, and said second language is XML.

25. The apparatus of claim 20, wherein the programming instructions further enable the apparatus to

- recognize a third code section with at least code statements of a third programming language;
- invoke a third code statement processing unit of the third programming language to process the third code section.

26. The apparatus of claim 25, wherein said third code section is embedded within said second code section, and said second code section is embedded within said first code section.

27. The apparatus of claim 25, wherein said first language is a directive language, said second language is Java and said third language is XML.

28. The apparatus of claim 20, wherein said programming instructions further enable the apparatus to

- recognize an invocation of a library function of a selected one of the first and the second programming language within the first code section,
- invoke the library function, and output the result of the invocation.

29. The apparatus of claim 28, wherein the library function is a selected one of an emit function for outputting execution results, a pop function for returning an element, and a push function for backing up an insertion point.

30. The apparatus of claim 20, wherein the said programming instructions are further designed to enable the apparatus to

- recognize a header section of a selected one of the first and the second programming language;
- recognize a directive statement within the header section, enumerating one or more data packages; and
- import the enumerated one or more data packages for use by code sections with at least code statements of the selected one of the first and the second programming language.

31. The apparatus of claim 20, wherein said programming instructions are further designed to enable the apparatus to

- recognize a header section of a selected one of the first and the second programming language;
- recognize a declare statement within the header section, enumerating one or more processing methods; and
- instantiate the enumerated one or more processing methods for use within code sections with at least code statements of the selected one of the first and the second programming language.

32. The apparatus of claim 20, wherein said programming instructions are further designed to enable the apparatus to

- recognize a header section of a selected one of the first and the second programming language;
- recognize a declare statement within the header section, enumerating one or more instance variables; and
- instantiate the enumerated one or more instance variables for use code sections with at least code statements of the selected one of the first and the second programming language.

33. An apparatus comprising:

- at least one storage medium having stored therein a plurality of programming instructions designed to enable the apparatus to
  - read a data processing representation having code sections with programming language statements of at least a first and a second programming language,
  - recognize a header section of a selected one of the first and the second programming language;
  - recognizing a directive statement within the header section, enumerating one or more data packages, and

import the enumerated one or more data packages for use code sections with at least code statements of the selected one of the first and the second programming language; and at least one processor coupled to the storage medium to execute the programming instructions.

34. The apparatus of claim 33, wherein said programming instructions are further designed to enable the apparatus to

recognize a declare statement within the header section, enumerating one or more processing methods, and

instantiate the enumerated one or more processing methods for use within code sections with at least code statements of the selected one of the first and the second programming language.

35. The apparatus of claim 33, wherein said programming instructions are further designed to enable the apparatus to

recognize a declare statement within the header section, enumerating one or more instance variables, and

instantiate the enumerated one or more instance variables for use within code sections with at least code statements of the selected one of the first and the second programming language.

36. An apparatus comprising:

at least one storage medium having stored therein a plurality of programming instructions designed to enable the apparatus to

read a data processing representation having code sections with code statements of at least a first and a second programming language,

recognize a header section of a selected one of the first and the second programming language,

recognize a first declare statement within the header section, enumerating one or more processing methods, and

instantiate the enumerated one or more processing methods for use within code sections with at least code statements of the selected one of the first and the second programming language; and at least one processor coupled to the storage medium to execute the programming instructions.

37. The apparatus of claim 36, wherein said programming instructions are further designed to enable the apparatus to

recognize a second declare statement within the header section, enumerating one or more instance variables, and instantiate the enumerated one or more instance variables for use within code sections with at least code statements of the selected one of the first and the second programming language.

38. An apparatus comprising:

at least one storage medium having stored therein a plurality of programming instructions designed to enable the apparatus to

read a data processing representation having code sections with code statements of at least a first and a second programming language,

recognize a header section of a selected one of the first and the second programming language,

recognize a declare statement within the header section, enumerating one or more instance variables,

instantiate the enumerated one or more instance variables for use within code sections with at least code statements of the selected one of the first and the second programming language; and

at least one processor coupled to the storage medium to execute the programming instructions.